

ANKARA UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT
COM1001 Computer Programming I
Fall 2023-24
PA3

Due Date: 29.12.2023 23:59

Task: Given a date, item names and their prices according to days of one week in a store, please find the day for the date according to Zeller's Algorithm, provide the prices of the queried items for the day and total price if there is a discount or not.

Detailed Specifications:

- You should find the day for the given Gregorian date according to Zeller's Algorithm.
- Assume that each item appears only once in the store list.
- Item names will be queried from the standard input using capital letters only.
- If the query item name is not in the given list, your program should print **0** to the screen.
- If the given date is 1 January, you should apply %20 new year discount to the total price.
- If the day for the given date is a weekend day (Saturday or Sunday), you should apply %10 weekend discount to the total price.
- You should print out the total price of the queried items with discount expression in parantheses. The **fractional part of the total price** should be printed on the screen with **only one digit**.
- The list of item names and their prices are separated from the query item names with the word: **start**
- The query items are terminated with the word: **end**

Input format: [S] stands for a Space/Tab character

<The_Given_Date>

<Item_Names>[S]<Day1>[S]<Day2>[S] <Day3>[S] ... [S]<Day7>

<item1_name>[S]< item1_price1>[S]< item1_price2>[S] ... [S]< item1_price7>

<item2_name>[S]< item2_price1>[S]< item2_price2>[S] ... [S]< item2_price7>

...

<itemN_name>[S]< itemN_price1>[S]< itemN_price2>[S] ... [S]< itemN_price7>

start

< itemX_name>

< itemY_name>

< itemZ_name>

...

end

Output Format:

<The_Day>

Zeller's Algorithm returns

<itemX_name>[S]<itemX_price_for_the_day>

<itemY_name>[S]<itemY_price_for_the_day>

<itemZ_name>[S]0

does not exists!

<Total:>[S]<Total_Price>[S]<(Discount_Expression)>

...

Zeller's Algorithm

From Wikipedia, the free encyclopedia

Zeller's congruence is an [algorithm](#) devised by [Christian Zeller](#) in the 19th century to [calculate the day of the week](#) for any [Julian](#) or [Gregorian calendar](#) date. It can be considered to be based on the conversion between [Julian day](#) and the calendar date.

Formula [\[edit \]](#)

For the Gregorian calendar, Zeller's congruence is

$$h = \left(q + \left\lfloor \frac{13(m+1)}{5} \right\rfloor + K + \left\lfloor \frac{K}{4} \right\rfloor + \left\lfloor \frac{J}{4} \right\rfloor - 2J \right) \bmod 7,$$

for the Julian calendar it is

$$h = \left(q + \left\lfloor \frac{13(m+1)}{5} \right\rfloor + K + \left\lfloor \frac{K}{4} \right\rfloor + 5 - J \right) \bmod 7,$$

where

- h is the day of the week (0 = Saturday, 1 = Sunday, 2 = Monday, ..., 6 = Friday)
- q is the day of the month
- m is the month (3 = March, 4 = April, 5 = May, ..., 14 = February)
- K the year of the century ($year \bmod 100$).
- J is the [zero-based](#) century (actually $\lfloor year/100 \rfloor$) For example, the zero-based centuries for 1995 and 2000 are 19 and 20 respectively (not to be confused with the common ordinal century enumeration which indicates 20th for both cases).
- $\lfloor \dots \rfloor$ is the [floor function](#) or integer part
- mod is the [modulo operation](#) or remainder after division

In this algorithm January and February are counted as months 13 and 14 of the previous year. E.g. if it is 2 February 2010, the algorithm counts the date as the second day of the fourteenth month of 2009 (02/14/2009 in DD/MM/YYYY format)

CHECK OUT THE SHARED INPUT AND OUTPUT FILES!

Submission:

- 1- Name your Python source file as student_id.py; replace student_id using your student id number.
- 2- Upload your python file using the interface provided in e-kampus course page.

Compiling Process:

//normal compile process

python3.8 yourfilename.py

//to use .txt file as input

python3.8 yourfilename.py<input.txt

// to use .txt file as input and to print the results to .txt file

python3.8 yourfilename.py<input.txt>output.txt

// to compare two files

diff -w filename1 filename2